

Sujet Examen Framework JS

Modalités de rendu

Vous avez toute la dernière journée de cours (17/05/2023) pour entamer le projet et me poser vos questions.

Vous aurez ensuite jusqu'au vendredi 16/06/2023 à 20h00 (heure d'envoi du mail faisant foi) pour m'avoir transmis par mail (mari.doucet.ext@eduservices.org) les liens de vos rendus :

Format du mail

En objet : [MDS M1 JS] Prénom Nom Rendu

Corps du mail :

- Le lien vers le repository Git public de votre API
- Le lien vers le repository Git public de votre Front

Barème de notation

Chacune des deux parties de l'examen (API et Front) est notée sur 20 et je fais ensuite la moyenne des deux notes.

Barème commun aux deux parties :

Élément du barème	Nombre de points maximum
Justification des choix	2 points
Notice d'installation	3 points
Documentation	2 points
Code commenté	3 points
Ça tourne en local sans erreurs	3 points
Ça tourne en local sans warnings injustifiés	2 points
Points bonus en réserve	5 points
Total	20 points

Les points bonus sont là pour permettre de la souplesse dans le barème, pour éventuellement rattraper certaines erreurs ou certains oublis, ou pour récompenser des bonus.

Barème spécifique à l'API :

Élément du barème	Nombre de points maximum
CRUD des Utilisateurs	1
CRUD des Dépenses	1
CRUD des Catégories de Dépense	1
Utilisation des méthodes HTTP appropriées	2
Total	5

Barème spécifique au Front

Élément du barème	Nombre de points maximum
Notice d'utilisation	1
Sémantique HTML respectée	1
Validation des formulaires	1
Messages d'erreurs explicites	2
Total	5

Le sujet

Vous allez devoir créer un système qui permet de faire une partie des fonctionnalités d'une application comme Tricount mais en version réduite.

Votre système permettra de gérer les dépenses d'une collocation :

Vous avez des utilisateurs, des dépenses et des catégories de dépense.

Les dépenses

Pour chaque dépense nous devons pouvoir savoir qui a payé, combien iel a payé, quand iel a payé, pour qui iel a payé et de quel catégorie de dépense il s'agit.

Les utilisateurs

Les utilisateurs ont un nom, un prénom et un email (unique). L'authentification n'est pas obligatoire sur le projet vous n'avez donc pas à gérer de système de connexion / mot de passe.

Les catégories de dépense

Les catégories de dépense ont un nom.

Les fonctionnalités minimum

Enregistrer un utilisateur

Les utilisateurs doivent pouvoir être créés.

Enregistrer une dépense

Un utilisateur doit pouvoir enregistrer une dépense.

Lister les dépenses

Les utilisateurs doivent pouvoir voir la liste complète des dépenses classées par date de la plus ancienne à la plus récente.

Créer une catégorie de dépense

Les utilisateurs doivent pouvoir créer une nouvelle catégorie de dépense si celle-ci n'existe pas

La balance comptable

Les utilisateurs doivent pouvoir savoir pour chacun combien il a payé et combien lui est dû.

Les technologies

Vous pouvez utiliser n'importe quelle framework / technologie JavaScript tant pour votre API que pour votre Front.

La documentation

Toute votre documentation doit être au format Markdown.

README

Dans votre README.md vous ferez figurer la justification de vos choix techniques, la liste de vos dépendances potentielles et une notice qui décrit pas par pas l'installation en local (et ce sur chacun de vos deux repos) ainsi que toute information supplémentaire qui vous paraît pertinente (par exemple si certains warning persistent et ne sont pas de votre fait). Vous pouvez la rédiger en français ou en anglais.

HOWTO

Dans le repo de votre Front vous ferez figurer un fichier HOWTO.md qui explique comment réaliser les principales tâches sur votre application : création d'une dépense, ajout d'un utilisateur, etc etc.